

2

Getting Started With Workflow Development

In this chapter, we will start to build a process from a blank canvas. In each subsequent chapter, we will continue to add different elements to the overall process to produce a fully working workflow which includes examples of how to include the vast majority of business requirements.

Objectives

By the end of this section, you should be able to

- Create a new workflow using the Quick Start wizard.
- Create a new workflow manually.
- Define the following workflow terms
 - Item Type
 - Process
 - Item Key
- Use the validate tool, and understand some of its limitations.
- Create and set Start and End nodes.
- Use the “Standard” workflow item type.

Starting Workflow Development

I've made an assumption at this stage that you've already got Workflow installed somewhere. This can be within Oracle Applications, the version that comes with Oracle Collaboration Suite, the version that comes with the Oracle Database, or the version that comes with the Application Server. The slight variations in the versions shouldn't make that much difference to following the process through the book - however, the server version that you are running should be **AT LEAST** version 2.6.2. The majority of the examples I'll be using in the book will be taken from a standalone installation of Workflow 2.6.2 running in a *9i* database.

When you start up the Workflow Builder, the first thing that you'll probably notice is that it's empty, and you don't get a pop-up “Getting Started” message like you do in Reports or Forms. All you get is one empty window inside another one - no navigator bar to help you on your way or anything as easy as that. So let's get straight into it and create a workflow. . .

Creating a Workflow

There are two different ways to get started with a new workflow - either manually start from scratch, or use the Quick Start wizard. By far the easiest (as the name suggests!) is to use the wizard, as this will create an empty shell of a workflow for you.

Using the Quick Start Wizard

The Quick Start Wizard can be started by

- Clicking on the light bulb icon on the Navigator bar shown in Figure 2-1.



Figure 2-1. Workflow Builder Navigator Bar

- Using the menu (File >> Quick Start Wizard)
- Or using the keyboard shortcut `Ctrl-Q`.

Whichever way you choose to open the wizard, you are presented with the pop-up box shown in Figure 2-2 prompting you to enter some information, all of which is required.

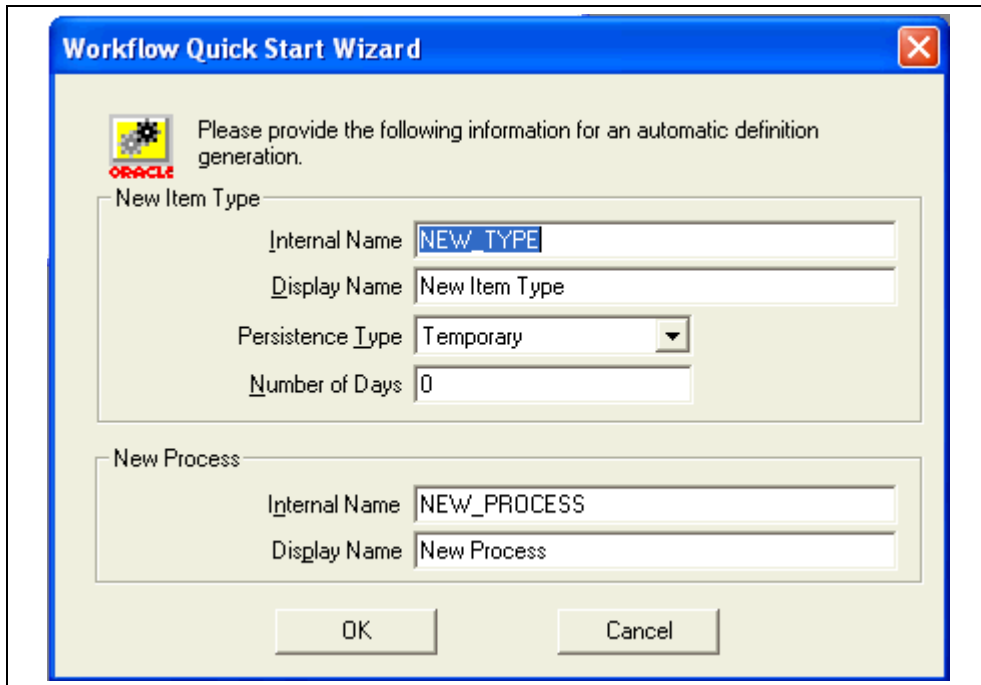


Figure 2-2. Workflow Quick Start Wizard

This introduces us to some new terms that are absolutely vital for workflow development:

Item Type

An item type can be defined as a collection of associated objects that comprise the workflow. Each item type can contain a number of different objects, which we'll define in more detail as we go through the book.

Process

A process can be defined as a workflow implementation of a flow of activities which need to be performed before the process can complete.

Display Name

The name of an object within the workflow, which is visible to the user.

Internal Name

The internal name of an object is the name that will always be referred to within the workflow. This **MUST** be unique within the item type. Whenever the object is referred to, this is the name that will be used rather than the display name (see above). Display and Internal names are described in more detail in the FAQ section.

Persistence Type

Oracle Workflow allows the developer to determine how long each workflow should remain in the database after they (and any linked workflows) complete before they become eligible for purging. The different persistence types are

Permanent

The standard API will never remove the workflow.

Temporary

Remain ineligible for purging until *n* complete days have passed.

Synchronous

Do not persist the workflow in the database, so no run-time data is stored.

Having opened the Quick Start Wizard, we'll go ahead and create the item type for using for creating the Insurance Claim workflow. As I've said above, once you pick an internal name it's difficult to change it, so try and pick something that you find useful or meaningful – since the internal name of an item type is limited to only eight characters, it is normally difficult to come up with something that is meaningful to everyone!

The internal name that I will be using for my one is `INSCLAIM` and I've called the process at this stage `MAIN_PROCESS`. Whenever I am creating a brand new workflow, I always try to call the process which is going to be the main driver something like `'MAIN'` so that it is immediately clear where the starting point is going to be. Figure 2-3 shows the names that I have chosen for my Item Type and Process.

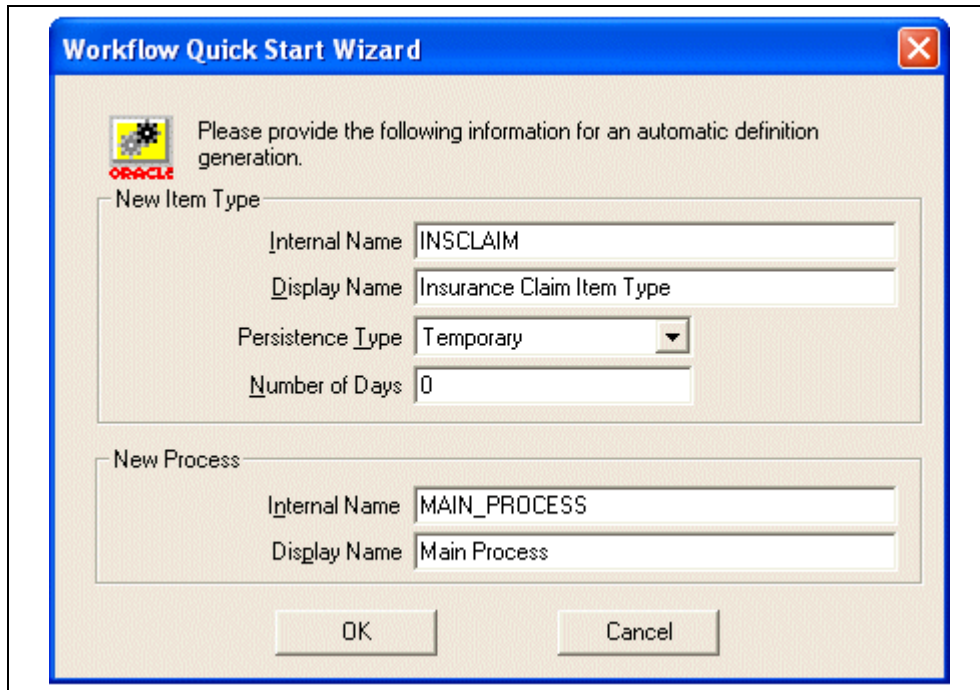


Figure 2-3. My Completed Quick Start Wizard

Before clicking on the OK button (don't worry if you have already done it!), move the window to one side and have a look at the navigator pane. When the wizard is started, the Workflow Builder loads a template file which it will use for creating the new item type, which is shown in Figure 2-4.

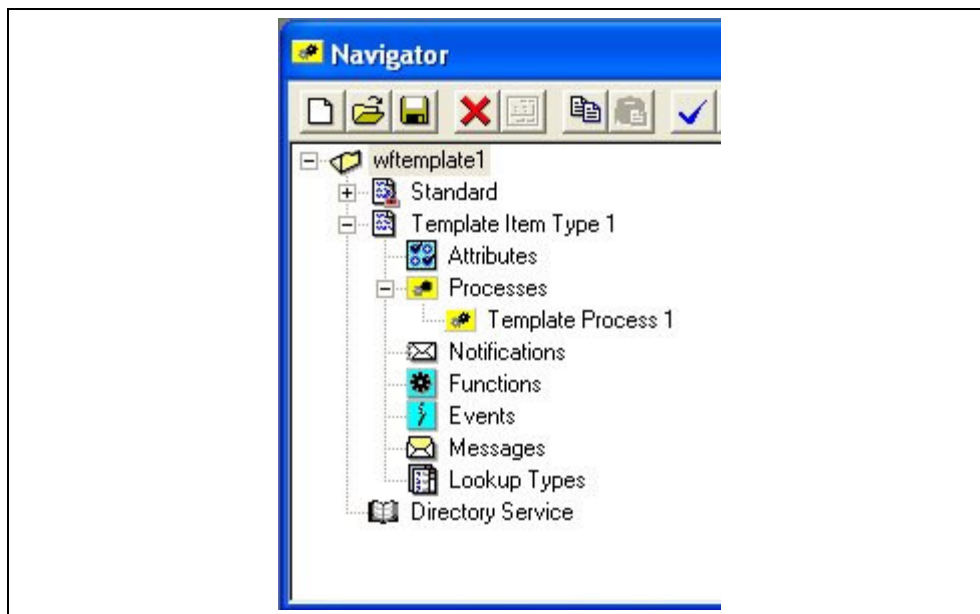


Figure 2-4. The Default Workflow Creation Template

As we continue developing the workflow, we'll be adding to every node that is there, and I'll explain each node when we get to that particular section.

Now click on the OK button to create your new item type and process.

Creating a Workflow Manually

The first thing that I should highlight about creating a Workflow manually from scratch is that it's much easier to use the Wizard described above. My advice is that whenever you are starting a completely new Workflow, use the Wizard. Once there is a basic framework there, it is a lot easier to build on than having to create it all yourself, but here's how to do it.

Firstly, create a new "Store". This is going to become the workflow definition file when you save it, and will contain item types as you create them. To create a new store, again there are three different mechanisms:

- Click the blank page icon on the navigator bar;
- Double-click on the folder icon in the navigator window;
- Use the keyboard shortcut Ctrl-N.

Unlike using the Wizard, the result is particularly unspectacular – a new folder appears with a name starting with "Untitled", and so the rest of the Workflow needs to be manually added.

Since we will need to have an item type to contain all the processes, you should create one either by right clicking on the folder icon and selecting "New Item Type..." or from the menu choose "Edit -> New -> Item Type..." which will open a new property palette window, as shown in Figure 2-5.

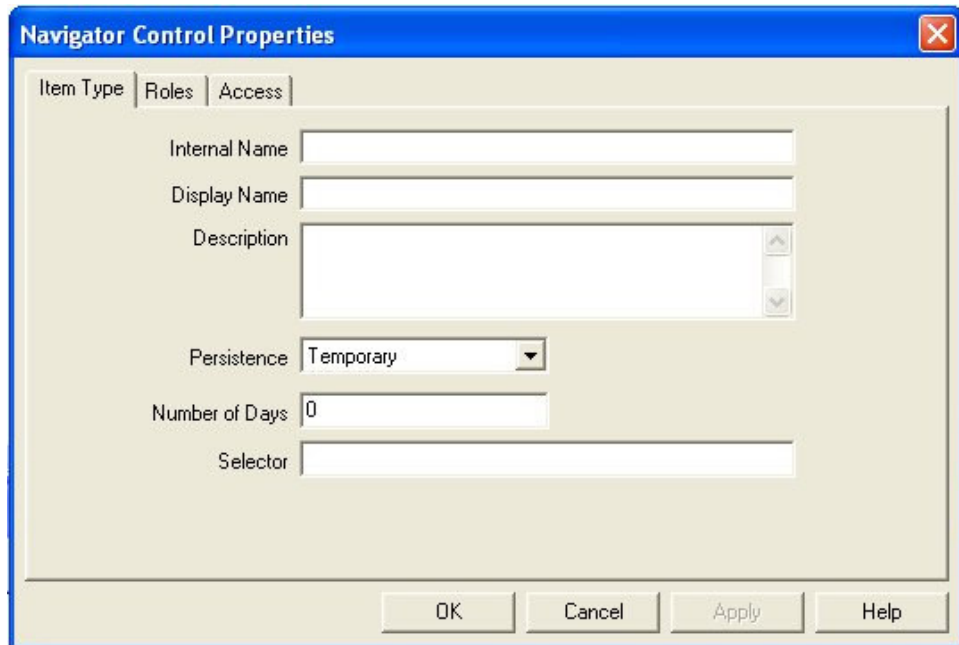


Figure 2-5. New Item Type Property Palette

The information that you can enter here for your new item type is the same as for creating an item type via the Wizard, plus a little bit more – the Selector field, as well as those on the Roles and Access tabs can be ignored for the time being. The mandatory fields are the internal and display names, so if you are creating an item type manually you should enter these and then click on the OK button.

One of the most important reasons for using the Wizard for creating your workflow store initially is that the wizard will automatically import a seeded Workflow item type called “Standard”. The Standard item type contains common code and lookups which can be used in your custom workflow – the most common of these functions are the “Start” and “End” functions. If you are creating your workflow manually (and hopefully by this point you’ll have realized why you should use the Wizard instead!!), you need to open the file (which is stored in the `Workflow Home\wf\DATA\US\` directory and is called `WFSTD.wft`) and then drag the “Standard” item type into your Untitled store.


Note on the Standard item type

Whilst it is not necessary to import the Standard item type, a number of useful activities and standard lookups are provided within the definition. These standard modules can greatly enhance how easy it is to develop your Workflow definitions, and so for the rest of the book I will be making the assumption that you have imported the Standard item type.

If the Standard item type is changed, this will affect all workflows in the database which rely on the item type. This is one reason why the Standard item type should **NEVER** be changed.

If you are in an Oracle Applications environment, the version of the Standard item type that is shipped in the database is normally different from the version that you will see here – this is because various development teams have added different pieces of code to the Standard item type during development, which have inadvertently been added to the shipped product.

The penultimate step in manually creating your workflow shell is to create the process, which can be done in any of four ways:

- Right-click on the “Process” node for your item type and choose “New Process”;
- Double-click on the “Process” node icon;
- From the menu choose “Edit -> New -> Process...”;
- Click on the “New Process” icon on the navigator bar ().

A new window will open, prompting you to supply the process details. Again, the only mandatory fields are the internal and display names, so you should fill those in and click the OK button to continue.

Finally, you are ready to begin defining your process! In order to get to the same stage as that provided by using the Wizard, you should expand the Standard item type, and also the “Functions” node within the item type. Click and drag the “Start” and “End”

functions into the Workflow process, and the workflow that you have just created should be the same as if you had pressed **CTRL-Q** and invoked the Quick Start Wizard!

Saving your Workflow

Regardless of the method you used to create the workflow shell, it is important to save it at this stage. Unlike some Oracle tools, Workflow builder does not have any form of undo functionality, apart from abandoning changes and reloading a previous version of the file.

Once you have made changes to the workflow file that you are sure that you want or need to keep, I would save the definition to a flat file and then ensure that you keep that copy safe – one project I worked on had over a hundred old versions of the definition file, just in case!

When you save the definition, the Workflow builder will implicitly invoke the validation tool, which may report a few errors, particularly if you created the process manually. Even if there are errors reported, these will not stop you from saving the definition – it is only when the workflow process is executed in the database that any errors stop processing continuing. An example of a save validation error window is shown in Figure 2-6.

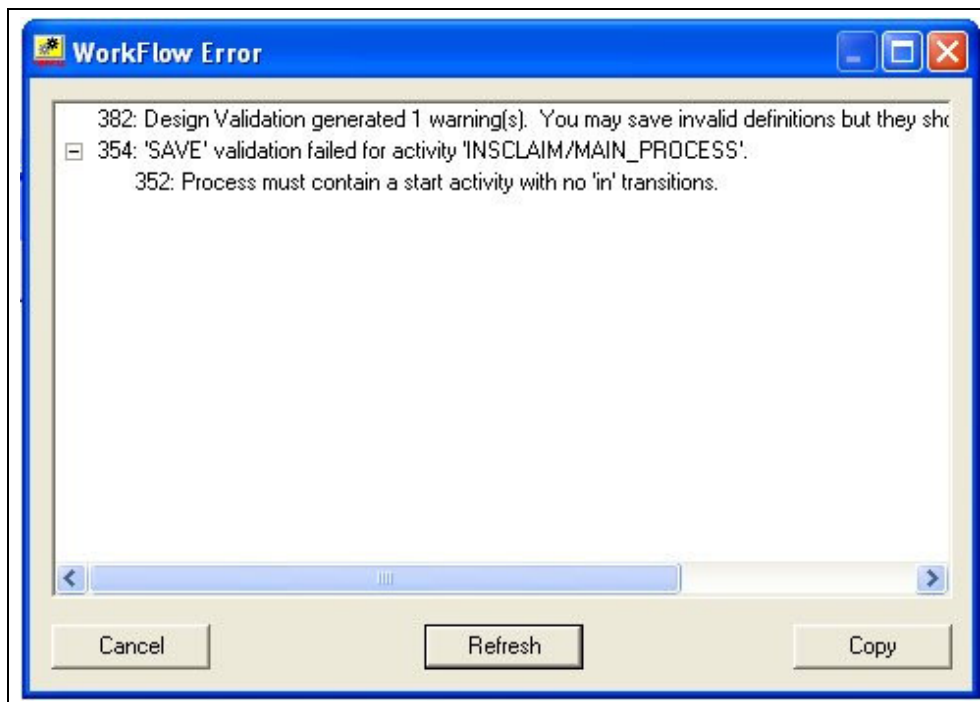


Figure 2-6. Save Validation Error Example

Validating Your Process

As I have just mentioned, whenever you try to save a process, the validation function will be automatically invoked. However you can manually call the validation tool by clicking on the Blue tick icon on the navigator toolbar.

If you created your workflow manually, you will have seen an error message displayed when you tried to save the workflow. If you used the Quick Start Wizard to create your workflow, then the process should have saved correctly. A manually created process is shown in Figure 2-7, and the process created by the Quick Start Wizard is shown in Figure 2-8.



Figure 2-7. Manually Created Process



Figure 2-8. Quick Start Wizard Created Process


HELP! *My process doesn't look like that!*

Your process should look like either of the diagrams above. This is the “end user” view – when a user looks at a process, this is what they will see, and so icons should be used to help users determine what the steps of the process are doing.


If your workflow looks like the one in figure 2-9 then don't worry – your workflow builder is just presenting you with the process in developer mode.



Figure 2-9. Viewing a Process in Developer Mode

You can switch between developer mode and the end user view by clicking on either of the two hammer icons (), choosing View -> Developer Mode from the menu, or by using the Ctrl-D keyboard shortcut.

When you change to Developer Mode, the text underneath the icons also changes. Developer mode will always display the type of activity (in this case using the standard icon for a function), and its instance label (START or END). You can change what text is shown by

choosing from the different icons on the right of the workflow diagram window (). In order from left to right, the options are:

- Show the instance label (from the Node -> Label tab for the activity)
- Show the internal name
- Show display name (generally the standard view when in user mode)
- Show comments (from Node -> Comment field for the activity)
- Show performers.

When viewing the diagram in user mode, I would normally choose to show the display name, since this is what is shown in the Workflow Monitor when a user searches for their process.

I will discuss the benefits of using developer mode in Part Two of this book.

True and False Starts

If you created your process using the Quick Start Wizard, then the Wizard will have created a process and included a Start and an End point, as shown above. The difference between this process and one created manually is that the Wizard has marked the activities as being a “true” Start and a “true” End. The difference between a “true” and a “false” start is that the icon in your process has a small green arrow in the bottom left hand corner for a “true” start. If the process does not have the arrow, then there is nothing that identifies to the Workflow Engine that this is the point that the process begins – just having an activity which you have called “Start” does not enable the Engine to recognize it as a starting point.

If the Start and End nodes are not correctly configured, you may see the following errors:

Process must contain a start activity with no ‘in’ transitions

This error message introduces us to some new terms:

Activity

A workflow Activity can be defined as a step in process, which indicates that some degree of processing must be performed. Within Oracle Workflow, each activity can be defined as a function (either in PL/SQL or Java), a notification, a business event or a sub-process.

Transition

A transition is the join between one activity and the next. Each activity can have zero or more transitions linking them to the subsequent activities in the process. The linking of the activities together with transitions forms your workflow process.

The error message should really be considered in two separate parts, so let's look at what it actually means.

1. The process must contain a start activity.

If you double-click on the Start icon within your diagram, a new dialogue box will open with a number of tabs on it. The first tab contains the important details for the activity, which we will examine in more detail when we start defining our own custom activities, including what code should be called when the activity is invoked. For the standard Start node, the WF_STANDARD.NOOP (pronounced No-Op) procedure will be called, which actually performs no actual processing. Each function must have some degree of code behind it, and so in order to use the standard Start node within the process, the NOOP code is called, since it will begin and end with no significant overhead.

In order to mark the activity as a "true" start node, open the "Node" tab and select "Start" from the drop down list for "Start/End", instead of the "Normal" value which is used as default. Once you OK the change from a standard node to a start node, the small green arrow will appear on the icon in the Workflow diagram.

2. The start node cannot have any 'in' transitions.

The second part of the error message is perhaps easier to explain. Since the node is being marked as a start, you should not have any previous processing which feeds into this node.

The exception to this is that it is possible to have multiple start nodes, including ones which appear in the middle of a process. Therefore, the workflow must have at least one Start node which does not have any inbound transitions, so that it is clear where the process should really start.

Although I have used different Business Events to trigger the starting of a new workflow process, I can honestly say that I have **NEVER** built or even seen a workflow which includes multiple Start nodes in this manner. Whilst it is possible to have multiple start points like this, it should be avoided at all costs, since there is no way for a developer, user or the Workflow Engine to determine which particular start point should initiate processing.

Process must contain at least one end activity

This error is similar to the "missing start node" error described above, since the activity which is being used as an End node within the process is still set to being a normal activity instead of an End. Once the node has been marked as an End, a small red arrow will appear in the bottom right hand corner of the icon.

You should change the function activity to be an End node, in the same way that you marked the Start node correctly.

End activities may not have transitions from them. These transitions would never be taken.

In the same way that a Start node should not have any in transitions, if you are marking an activity to be the End point of the process then it cannot have any processing after that point, and so you should not have any outbound transitions to further processing.

One key difference between the processing of an End node and a Start node is that it is perfectly normal to include multiple Ends within the process, either for ease of diagramming or because the process needs to return different values to the calling code. As soon as one End node has been reached, the process will stop completely. Consider the workflow process shown in Figure 2-10.

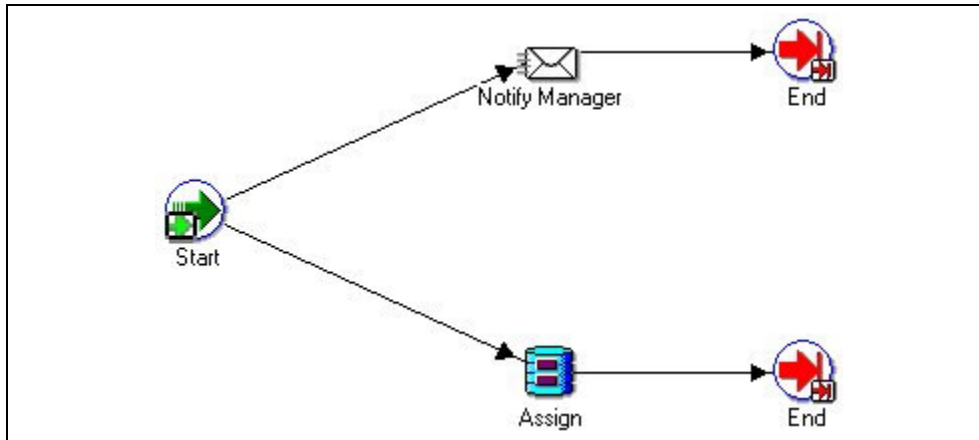


Figure 2-10. How not to use multiple end nodes

As soon as the Workflow Engine transitions to the top End node, any processing which might have happened on the parallel processing branch will stop. In this example, you cannot guarantee which activities will be processed – will the “Notify Manager” step be executed as well as the “Assign” activity? You should take care when using multiple End points that all required processing will have completed before one End point has been reached. The better way to model the process would be as shown in Figure 2-11.

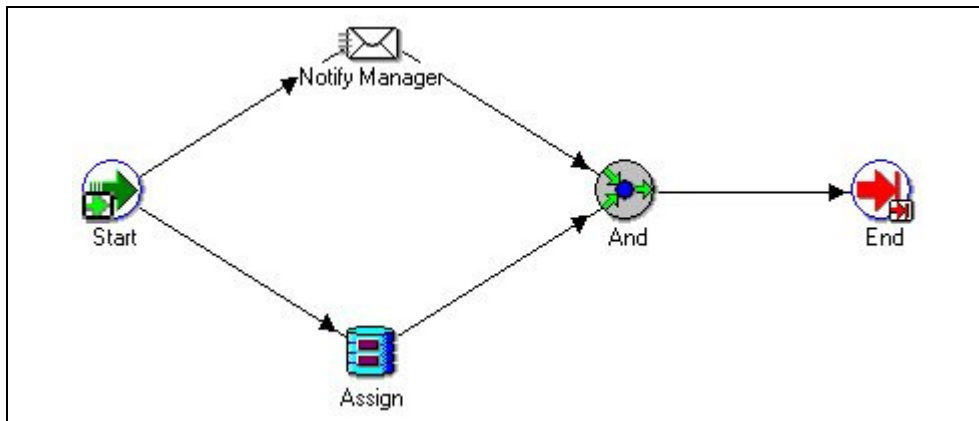


Figure 2-11. How to Ensure Parallel Branches Complete

Limitations of the Validation Tool

After setting the Start and End nodes within the process, this should resolve the errors which the Validation tool has reported, which should be confirmed by manually validating the definition.

One of the key problems with the Validation Tool is that at no stage does it actually check whether all the activities in the process connect! According to the validation tool, the process which you have created is valid, even though there is no link between the Start and End points!

So what does the validation tool actually check?

- The process has one or more Start nodes;
- The process has at least one Start node with no inbound transitions;
- The process has one or more End nodes;
- End nodes do not have any outbound transitions;
- Any notifications that are being sent have a recipient specified.

What key things does the validation tool NOT check?

- That there is a connection between a Start node and an End node;
- That any code for an activity exists, is valid and installed in the database.

This list seems to imply that the elements that *are* checked outweigh those that are not being checked, however the two things listed above are incredibly important. Of these two points, the fact that the Workflow builder does not connect to the database to verify the existence of any code is more of an issue than the first. No matter how complicated the process is, then it should be reasonably easy to spot whether the Start and End activities connect. This is particularly true if the workflow has been diagrammed so that it is easy to read and understand.

There is a fundamental reason that the Workflow builder does not connect to the database to validate whether the code exists – Workflow development can be split easily between multiple developers who may not be delivering the PL/SQL or Java components and the Workflow components at the same time. Additionally, Workflow can be started by a non-technical developer / business analyst and then passed to a technical developer who can complete the Workflow design from a technical perspective and also deliver the supporting PL/SQL or Java code.

On one of my early Workflow projects, where we had a split development team between the PL/SQL and Workflow components, the fact that the builder does not validate the existence of the PL/SQL in the database became problematic. At the end of seven days of testing, the workflow progressed to the last stop – at which point it errored because the PL/SQL developer had named the procedure including an underscore, and the Workflow developer had not! This, and the potential for typing errors in entering the name of the PL/SQL package and procedure, is not uncommon when building workflows.

Completing your Workflow

Although the validation tool is informing you that the workflow developed so far has been “Successfully validated”, it is obvious just from looking at the Workflow diagram on the screen that there is something fundamentally wrong with it – the two nodes don’t connect!

The last thing that we will be doing in this chapter will be completing the workflow process, so that although it is basic to say the least, it is something that we can look to run in the next chapter.

In order to create the link (or transition) from the start point to the end, you should firstly click on the Start node, and then draw the transition between the two activities using the **right** mouse button – which is completely different from your “normal” Windows functionality! When you release the mouse button, a line will appear between the two steps of the process, pointing from the Start to the End.

The process should look something like the one shown in Figure 2-12.

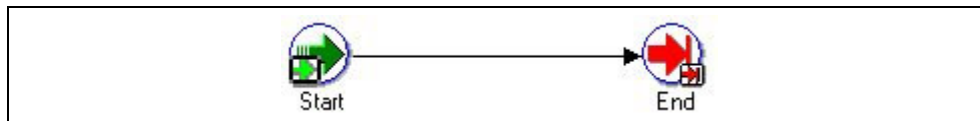


Figure 2-12. Insurance Claim Process So Far

If the arrow is pointing towards the Start node, then the process will fail validation, since you have a transition into a Start node and a transition out of the End node. Click on the link between the two activities, and delete it by either using the Delete key or right-click on the link and choose “Delete Selection” from the menu.

Once you have linked the two steps, you should validate your process to ensure that the basic requirements have been met before moving on to the next section – saving and deploying your Workflow definition to the database.

Deploying to the Database

Now that you have a completed process, the final step before we can move on to the next chapter and launch the process is to save the definition.

Firstly, you should save your workflow to a flat file by either

- Clicking on the floppy disk icon on the navigator bar;
- Using the Ctrl-S keyboard shortcut;
- Choosing “File > Save” or “File > Save As” menu options.

Whichever method you choose, you will be prompted with a dialog box like the one shown in Figure 2-13.



Figure 2-13. Save As Dialog Box

When the File radio button is selected, the top box is enabled and you the Workflow Builder will save the definition to your filing system. At this stage, you should save the definition as a file – whenever you are working with Workflow, you should **always** work with a flat file and then save to the database.

The reason that I recommend always working against a flat file rather than the database is to minimize any possibility of making changes which could impact the successful running of workflow processes which are already being executed by the Workflow Engine. For example, if you are making changes to parts of the Workflow which are not automatically version controlled by the Workflow system (such as messages) then this can impact on any running processes within the database.

Once the definition has saved successfully saved to a flat file, you should save it to the database. Invoke the Save As dialog window again, and this time select the Database radio button, which will enable the user, password, connect and effective fields.

You should provide the connection details for your Workflow system in the user, password and connect fields – if you are using standalone Workflow, then the username that you provide here will be the one which was used when installing Workflow into the database. If you are using Workflow embedded within the eBusiness Suite, then you should save to the database as the APPS user.

The Effective field is an optional field, which specifies at what point the process becomes active within the database. You can use this field to deploy a workflow to the database but only become useable at a certain point in the future – a customer of mine in the UK used the Effective field to manage their staggered implementation of the project. Due to a number of business requirements, the customer had a fixed implementation schedule of four different times a year. When the Workflow was ready for deployment, it was saved to the database with an Effective date set to the date at which the new version would go-

live. If you leave the Effective field blank, then the workflow will be useable immediately.

When the Workflow has saved to the database, the icon at the top of the navigator window will change from a folder icon and filename to a database icon and connection string, as shown in Figure 2-14.

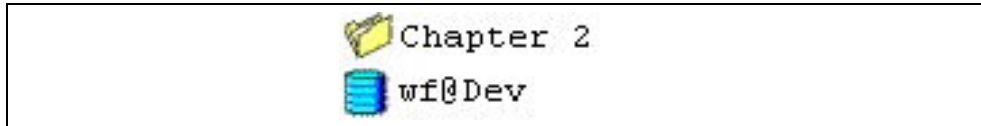


Figure 2-14. Different Save Icons in Workflow Builder

Having saved your Workflow definition to the database, let's move on to the next chapter and launch our new process.